Local Similarity Between Quotiented Ordered Trees

Pascal Ferraro, Aïda Ouangraoua, Laurent Tichit, Serge Dulucq

Local similarity between quotiented ordered trees

Pascal Ferraro [a,*] Aïda Ouangraoua [a] Laurent Tichit [a,1]
Serge Dulucq [a]

[a] *LaBRI - Université de Bordeaux 1*
351 Cours de la Libération, 33405 Talence Cedex, France

Abstract

In this paper we propose a dynamic programming algorithm to evaluate local similarity between ordered quotiented trees using a constrained edit scoring scheme. A quotiented tree is a tree defined with an additional equivalent relation on vertices and such that the quotient graph is also a tree. The core of the method relies on two adaptations of an algorithm proposed by Zhang et al. [1] for comparing ordered rooted trees. After some preliminary definitions and the description of this tree edit algorithm, we propose extensions to globally and locally compare two quotiented trees. This last method allows to find the region in each tree with the highest similarity. Algorithms are currently being used in genomic analysis to evaluate variability between RNA secondary structures.

Key words: Ordered labeled trees, local edition, quotiented graph, dynamic programming.

1 Introduction

The comparison of trees is an important operation applied in several fields, such as molecular biology [2,3], botany [4], pattern recognition [5], *etc.* To compute similarity between trees, edit distance metrics, initially introduced for string to string comparison problem [6], were first extended to compare

* Corresponding author.
Email addresses: pascal.ferraro@labri.fr (Pascal Ferraro), aida.ouangraoua@labri.fr (Aïda Ouangraoua), tichit@iml.univ-mrs.fr (Laurent Tichit), serge.dulucq@labri.fr (Serge Dulucq).
[1] Present address: IML - Université de la Méditerranée
Campus de Luminy, Case 907 - 13288 MARSEILLE Cedex 9

ordered trees [7,8] and then unordered trees [9] (see [10,11] for a review). A distance between two trees is thus computed as the minimum cost of a sequence of elementary operations that converts one tree into the other and minimizing the operation costs. In this article, we consider extensions of Zhang and Shasha [1] algorithm, that computes the distance by considering an optimal mapping between two trees. Note that Jiang et al. [12] have also introduced an alternative to mapping and tree edition called *alignment* of trees, but the notion of alignment won't be considered here.

To take into account the multiscale nature of different biological structures (*e.g.* plants [13], RNA [3,14]) Zhang and Shasha algorithm has been extended to compare quotiented ordered trees. A quotiented ordered tree [13] is a tree with an equivalence relation defined on the set of vertices, and such that the resulting quotient graph is also an ordered tree. A quotiented tree can thus be considered as an auto-similar structure represented by trees on two different scales. An equivalent problem has been solved by Ferraro and Godin [15] who proposed a constrained edit distance between *unordered* quotiented tree.

These distances allow the user to globally evaluate similarity between two trees or two quotiented trees. However, in many cases trees share only a limited region of similarity. We thus proposed extensions of Smith and Waterman algorithm [16] for evaluating local similarities between ordered trees and then to locally compare quotiented trees. The local edit score computation between two trees is an alternative to the basic global score computation algorithm which often gives different and yet sometimes more relevant results than the global approach when dealing with real biological data.

2 Definitions and notations

A *rooted tree* is a directed acyclic connected graph $T = (V, E)$ (V and E are respectively the set of vertices and edges) in which one of the vertex is distinguished from the others. The distinguished vertex is called the *root* of the tree (Figure 1.a). By extension, the particular graph $\theta = (\emptyset, \emptyset)$ is a tree and is called the *empty tree*.

Let $T = (V, E)$ be a rooted tree, $|T|$ represents the number of vertices of T. Let (v, w) be an edge of V, v is called the father of w and w is a child of v. A vertex that have no child is called a leaf. A vertex v is called an *ancestor* of a vertex w (and w is called a *descendant* of v) if there exists a sequence of vertices $(x_1, x_2, \ldots, x_n)$, called a path, such that $x_1 = v$ and $x_n = w$, and for each consecutive pair of vertices (x_i, x_{i+1}), x_{i+1} is a child of x_i. The ancestor relationship is a partial order relation denoted by $\preceq$.

A *complete* subtree (or simply a subtree) is a particular connected subgraph of a tree. Let $T = (V, E)$ be a tree, rooted in r, a subtree of T rooted in x

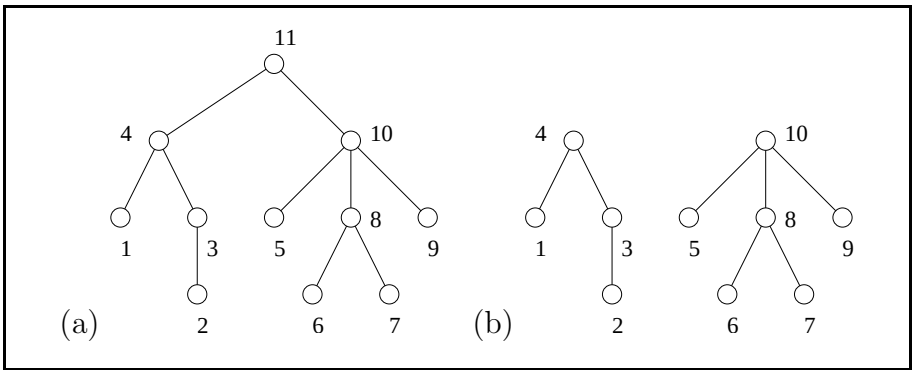

Fig. 1. (a) A tree with postfix order numbering and (b) the forest $F[1 \dots 10]$

is denoted by $T[x] = (V[x], E[x])$, where $V[x] = \{y \in V | x \preceq y\}$ and $E[x] = \{(u, v) \in E | u \in V[x] \text{ and } v \in V[x]\}$. A *partial* subtree is a connected subgraph of a subtree $T[x]$ which does not necessarily include all the descendants of x. For instance, in Figure 1.a, the sub graph of $T[10]$ made of vertices 10, 5 and 8 is a partial subtree.

A rooted tree is said *ordered* if the set of children of a given vertex are ordered. These are therefore trees for which the left-to-right order among the sibling vertices is significant (see [17] or [18]). In this paper, trees will be ordered according to the *postfix order*. The postfix order relationship on the vertices of an ordered rooted tree T rooted in r is obtained by visiting all the subtrees of T rooted on the children of r (in respect with the order on these children) and finally the root r. The postfix order relationship is a total order relation on vertices denoted by $\leq$. Moreover, vertices will purposely be identified with their postfix order index (Fig. 1a). The leftmost leaf descendant of the subtree rooted at vertex i according to the postfix order is denoted by $l(i)$ (for instance, in figure 1, $l(10) = 5$).

A *forest* is a directed graph whose connected components are ordered rooted trees. Referring to the notations of [3,19], let $x_1 < x_2 < \ldots < x_k$ be the vertices of $T[x_k]$, $F[x_1 \dots x_i]$ is the forest consisting in the subtrees of $T[x_k]$ restricted to the vertices $x_1, x_2, \ldots, x_i$. Particularly, $F[x_1 \dots x_k]$ is the whole tree $T[x_k]$. By convention, if $x_k < x_1$ then $F[x_1 \dots x_k]$ represents θ the empty tree.

A *quotiented tree* is a 3-tuple $Q = (T, W, \pi)$ where $T = (V, E)$ is a tree called the support of Q, W is a set of vertices and π a surjective application from V to W. For any vertex x in V, the vertex $\pi(x)$ is called the complex of x and reciprocally, x is a component of $\pi(x)$. $\pi^{-1}(X) = \{x \in V | \pi(x) = X\}$ denotes the set of components of a vertex X of W and if x is a vertex of V, $\pi^{-1}(\pi(x))$ is the set of components of $\pi(x)$. By convention, $\pi^{-1}(X)$ is identified with the subtree of T made of vertices in $\pi^{-1}(X)$. The function π induces a partition π_Q on V: $\pi_Q = \{\pi^{-1}(X) | X \in W\}$. The quotient graph $\pi(T)$ associated with Q is (W, E_π) such that: $\forall (x, y) \in E, (\pi(x), \pi(y)) \in E_\pi \Leftrightarrow \pi(x) \neq \pi(y)$. By definition, in a quotiented tree graph, quotient graph and support graph are

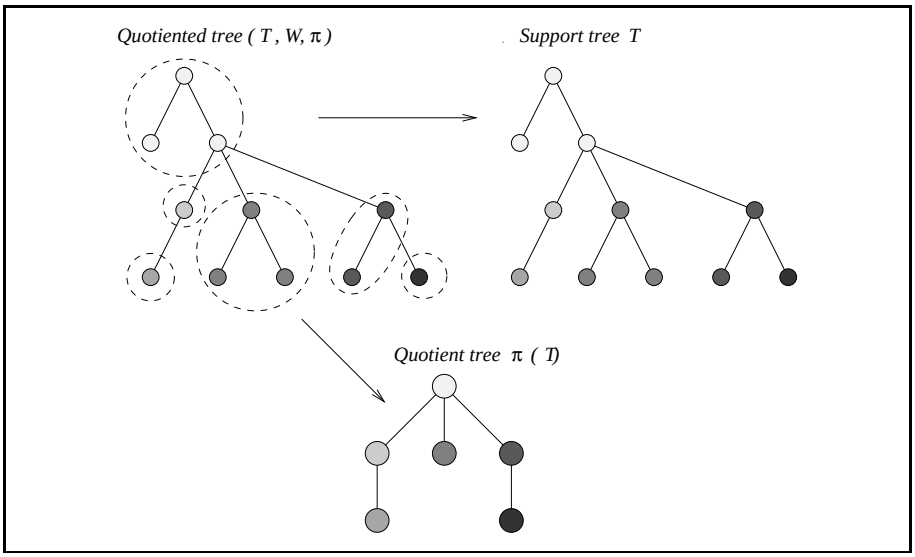

Fig. 2. A quotiented tree, its support T and its quotient $\pi(T)$

both trees (Fig. 2). A quotiented tree is said ordered if its quotient tree and support tree are both ordered. Thus, previous definitions and notations are still valid on quotiented trees. Let $X_1 < X_2 < \ldots < X_i$ be vertices of W, the quotiented subtree of Q rooted in X_1 is denoted by $Q(X_1)$. Referring to previous notation, $Q[X_1 \ldots X_i]$ is the quotiented forest consisting in the subtrees of $Q(X_i)$ restricted to the vertices $X_1, X_2, \ldots, X_i$.

In the following, we consider *labeled* ordered rooted (and eventually quotiented) trees and *labeled* forests. Each vertex of a tree $T = (V, E)$ or a forest is labeled by a symbol belonging to a finite set Σ of labels using a labeling function $\alpha : V \to \Sigma$. We will consider an edit score function on this set of labels. The score function s assigns a real number $s(a, b)$ to each pairs of labels (a, b) in $\Sigma \cup \{\lambda\}$ where λ represents the empty symbol ($s(a, \lambda)$ is the score of the deletion of symbol a in Σ and $s(\lambda, b)$ is the score of the insertion of b) such that:

$$\begin{aligned} s(a, a) &> 0 \quad \forall a \in \Sigma, \\ s(a, b) &< 0 \quad \forall a \neq b, a, b \in \Sigma \cup \{\lambda\}. \end{aligned}$$

This means that the score between two symbols a and b become higher with their similarity. Moreover, in the following, we constraint the score function to respect the dual of the triangle inequality for distance:

$$s(a, b) \geq s(a, c) + s(c, b), \forall a, b, c \in \Sigma \cup \{\lambda\}.$$

Usually, in comparison tree problems, labels and vertices are identified. Here, we will make the distinction between these both notions. So, for an edit operation on a tree consisting to the "transformation" of a vertex x in a vertex y (*transformation* means substitution, deletion or insertion), the resulting score is denoted by $s(\alpha(x), \alpha(y))$.

3 Global similarity

3.1 Global comparison between ordered trees

A considerable amount of works has been done on ordered tree comparison. Among various tree metrics, Taï [8] and Selkow [7] proposed an edit distance metric between ordered rooted trees based on the generalization of string comparison defined by Wagner and Fisher [6].

The tree-to-tree correction problem consists in determining the distance between two trees measured by the optimal sequence of edit operations needed to transform one tree into the other. Following Wagner and Fisher original definitions on sequences, three edit operations are used: *substituting* a vertex x into a vertex y means changing the label of x into the label of y, *deleting* a vertex x means making the children of x become a new children of the father of x and then removing x, *inserting* a vertex y means that y becomes the child of a vertex x and a subset of consecutive children (relatively to their order) of x becomes the set of children of y.

Let e be an edit operation, a score σ is assigned to each edit operation as follows: if e substitutes x into y then $\sigma(e) = s(\alpha(x), \alpha(y))$, if e deletes x then $\sigma(e) = s(\alpha(x), \lambda)$ and if e inserts the vertex y then $\sigma(e) = s(\lambda, \alpha(y))$. The score σ is extended to a sequence of edit operation $E = (e_1, e_2, \ldots, e_n)$ by letting $\sigma(E) = \sum_{i=1}^{n} \sigma(e_i)$. This makes it possible to define a similarity $S(T_1, T_2)$ between trees T_1 and T_2 as the maximum score of edit operation sequences transforming T_1 into T_2, namely

$$S(T_1, T_2) = \max_{E \in \mathcal{E}} \{\sigma(E)\},$$

where $\mathcal{E}$ represents the set of sequences of edit operations transforming T_1 into T_2. Likewise, we can extend this notion to the similarity between forests $S(F_1, F_2)$.

Zhang et al. [1] proposed a general recurrence formula for computing similarity between ordered forest. Let F_1 and F_2 be two ordered forests and let x_1, y_1 and x_2, y_2 be vertices of F_1 and F_2 respectively.

$$S(F_1[x_1 \ldots y_1], F_2[x_2 \ldots y_2]) = max \begin{cases} S(F_1[x_1 \ldots y_1 - 1], F_2[x_2 \ldots y_2]) + s(\alpha(y_1), \lambda) \\ S(F_1[x_1 \ldots y_1], F_2[x_2 \ldots y_2 - 1]) + s(\lambda, \alpha(y_2)) \\ S(F_1[x_1 \ldots l(y_1) - 1], F_2[x_2 \ldots l(y_2) - 1]) \\ + \; S(F_1[l(y_1) \ldots y_1 - 1], F_2[l(y_2) \ldots y_2 - 1]) \\ \quad + s(\alpha(y_1), \alpha(y_2)) \end{cases} \tag{1}$$

Note, if y_1 (resp. y_2) is an ancestor of x_1 (resp. x_2), then $F_1[x_1 \ldots y_1]$ (resp. $F_1[x_2 \ldots y_2]$) is a tree and $F_1[x_1 \ldots l(y_1) - 1]$ (resp. $F_2[x_2 \ldots l(y_2) - 1]$) is the

empty tree.

3.2 Global comparison between quotiented trees

Ferraro and Godin [15] have recently introduced an edit distance between unordered quotiented trees based on a comparison of support graph and edit operations that preserves equivalence relations. We propose here a symmetric approach by comparing quotiented trees at the more macroscopic scale. Basically, quotiented trees refer to trees whose nodes are also trees. Edit score related to quotient vertices is thus defined as an edit score computation between the support subtrees of these vertices.

Let $Q_1 = (T_1, W_1, \pi_1)$ and $Q_2 = (T_2, W_2, \pi_2)$ be two quotiented trees (if no confusion is possible π_1 and π_2 are denoted by π). Let e be an edit operation, the score σ_Q assigned to each edit operation is defined as follow:

- if e is a substitution of X_1 into X_2: $\sigma_Q(e) = S(\pi^{-1}(X_1), \pi^{-1}(X_2))$,
- if e is a deletion of X_1: $\sigma_Q(e) = S(\pi^{-1}(X_1), \theta)$,
- if e is an insertion of X_2: $\sigma_Q(e) = S(\theta, \pi^{-1}(X_2))$.

Like previously, σ_Q is extended to define the cost of a sequence of edit operations E from $\pi(T_1)$ to $\pi(T_2)$ by letting $S_Q(\pi(T_1), \pi(T_2)) = \sigma_Q(E) = \sum_{i=1}^{n} \sigma_Q(e_i)$.

A score between quotiented trees is then defined by the following optimization problem:

Problem 1 *Let Q_1 and Q_2 be two quotiented trees, find $\sigma_Q(E)$ maximum, such that E is a sequence of edit operation that transforms $\pi(T_1)$ into $\pi(T_2)$, namely:*

$$S_Q(Q_1, Q_2) = \max_{E \in \mathcal{E}_Q} \{\sigma_Q(E)\},$$

where $\mathcal{E}_Q$ represents the set of sequences of edit operations transforming $\pi(T_1)$ into $\pi(T_2)$.

Let $Q_1 = (T_1, W_1, \pi)$ and $Q_2 = (T_2, W_2, \pi)$ be both quotiented trees and let X_1, Y_1 and X_2, Y_2 be respectively vertices of $\pi(T_1)$ (*ie.* W_1) and $\pi(T_2)$ (*ie.* W_2), we can then deduce from equation 1 the following recurrence formula:

$$S_Q(Q_1[X_1 \dots Y_1], Q_2[X_2 \dots Y_2]) = max \begin{cases} S_Q(Q_1[X_1 \dots Y_1 - 1], Q_2[X_2 \dots Y_2]) + S(\pi^{-1}(Y_1), \theta) \\ S_Q(Q_1[X_1 \dots Y_1], Q_2[X_2 \dots Y_2 - 1]) + S(\theta, \pi^{-1}(Y_2)) \\ S_Q(Q_1[X_1 \dots l(Y_1) - 1], Q_2[X_2 \dots l(Y_2) - 1]) \\ \quad + S_Q(Q_1[l(Y_1) \dots Y_1], Q_2[l(Y_2) \dots Y_2 - 1]) \\ \quad + S(\pi^{-1}(Y_1), \pi^{-1}(Y_2)). \end{cases} \quad (2)$$

The main difference between this recursive relation and the computation of global edit score between trees lies in the computation of the score of the edit

operations between quotient vertices. There are computed as the edit score between the support subtrees corresponding to the quotient vertices.

4 Local similarity

In many cases trees share only a limited region of similarity. This may be a common domain or simply a short region of recognizable similarity. This case is dealt with by so-called *local mapping* in an algorithm developed by Smith and Waterman [16] to evaluate local similarity between strings. Local similarity aims at identifying the best pair of regions, one from each string, such that the optimal (global) similarity of these two regions is the best possible. This relies on a scoring scheme that maximizes a similarity score because otherwise an empty sequence of edit operations would always yield the smallest score. Naively, the algorithm to compute the local similarity would need to inspect every pair of regions and apply a global comparison algorithm to it. The decisive idea of Smith and Waterman was to find for any prefix of the sequences a suffix with a maximal score. We propose a generalization of this algorithm to evaluate local similarity between ordered and quotiented trees.

A first algorithm for finding similar regions in trees has been recently proposed by Höshmann et al. [20]. It build upon the tree alignment algorithm for ordered trees given by Jiang et al. [12]. This algorithm is used for evaluating local similarity in RNA secondary structures. However, since edition of trees and tree alignments are different concepts and lead to different algorithms, this method is not discussed in this paper. Note that in [21,22], Wang et al. solved a similar problem consisting of finding the largest approximately common substructures in ordered labeled trees. Given two trees T_1 and T_2 and an additional parameter δ, their algorithm determine the argest subtrees U_1 and U_2 of respectively T_1 and T_2 whose distance is at most δ. It is based on the computation of the global edit distance between two trees, and is therefore a minimization problem. Our variation does not use any additional parameter and thus cannot be solved as a minimization problem.

4.1 Local comparison between ordered trees

The computation of a local similarity allows to detect local conserved areas between both trees. The solution of such a problem is based on the notion of prefix mapping between trees.

Definition 2 *Let T be a tree rooted in r, any* partial *subtree of T rooted in r is called* a prefix of T *or* a T-prefix*. By convention, the empty tree θ is a T-prefix.*

Note that a particular prefix of T rooted in r is $T[r]$ itself. Let T_1 and T_2 be two trees and let x_1 and x_2 be two vertices of T_1 and T_2, the set of $T_1[x_1]$-prefixes and $T_2[x_2]$-prefixes are respectively denoted by $\mathcal{T}_1[x_1]$ and $\mathcal{T}_2[x_2]$.

A similar definition can be proposed for a forest:

Definition 3 *Let F be a forest made of n trees $T_1, \ldots, T_n$ respectively rooted in $r_1, r_2, \ldots, r_n$. A* F-prefix *is a sub-forest of F made of any prefixes of $T_1, \ldots, T_n$.*

The local prefix mapping problem for a given pair x_1, x_2 of vertices is to find a (possibly empty) prefix ρ_1 of $T_1[x_1]$ and a (possibly empty) prefix ρ_2 of $T_2[x_2]$ (Figure 3.a) such that the score of the optimal sequence of edit operations transforming ρ_1 into ρ_2 is the maximum over all scores of sequences of edit operations between prefixes of $T_1[x_1]$ and $T_2[x_2]$.

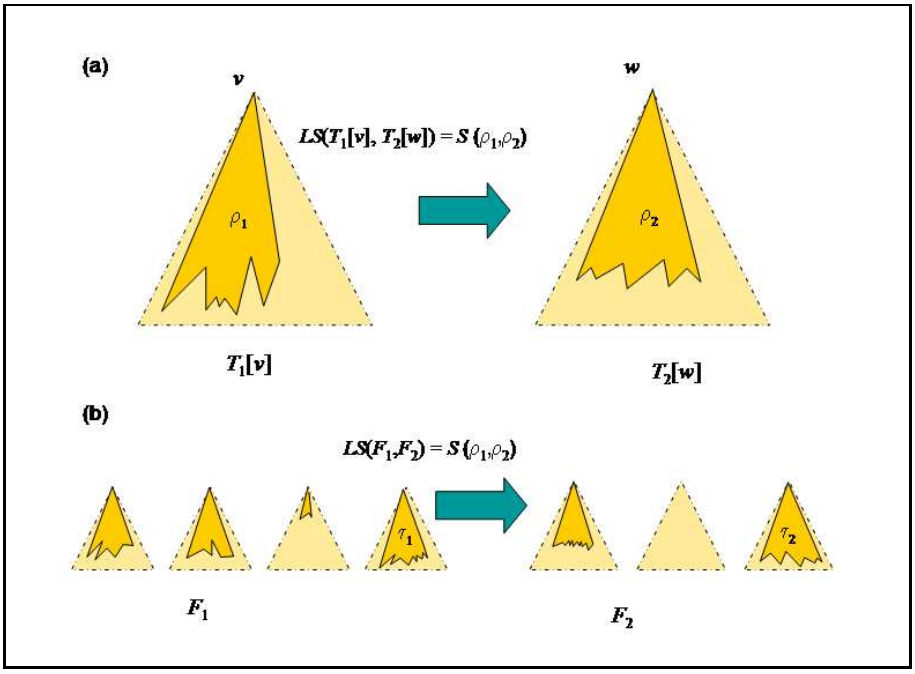

Fig. 3. Local prefix definition for (a) two trees and (b) two forests

The score of the sequence solving the optimal local prefix mapping problem (called local score) for a given pair x_1, x_2 of vertices is denoted by $LS(T_1[x_1], T_2[x_2])$:

$$LS(T_1[x_1], T_2[x_2]) = max\{S(\rho_1, \rho_2), (\rho_1, \rho_2) \in \mathcal{T}_1[x_1] \times \mathcal{T}_2[x_2]\}.$$

Note that a local prefix problem between two forests $F_1[x_1 \ldots y_1]$ and $F_2[x_2 \ldots y_2]$ is similarly defined as:

$$LS(F_1[x_1 \ldots y_1], F_2[x_2 \ldots y_2]) = max\{S_F(\rho_1, \rho_2), (\rho_1, \rho_2) \in \mathcal{F}_1[x_1 \ldots y_1] \times \mathcal{F}_2[x_2 \ldots y_2]\}.$$

where $\mathcal{F}_1[x_1 \ldots y_1]$ and $\mathcal{F}_2[x_2 \ldots y_2]$ represent respectively the set of $F_1[x_1 \ldots y_1]$-prefixes and $F_2[x_2 \ldots y_2]$-prefixes (Fig. 3.b).

Local similarity between two trees is then defined as the score of the best pair of local prefixes in trees T_1 and T_2:

Theorem 4

$$LS(T_1, T_2) = max\{LS(T_1[x_1], T_2[x_2]), (x_1, x_2) \in V_1 \times V_2\}.$$

So, in order to evaluate local similarity, the algorithm needs to find maximum similarity between prefixes of $T_1[x_1]$ and $T_2[x_2]$, for any pair of vertices (x_1, x_2) of $V_1 \times V_2$, and then to determine the best pair of vertices x_1^{Max}, x_2^{Max} of T_1 and T_2.

4.2 Case of trees

Let T_1 and T_2 be two trees respectively rooted in x_1 and x_2 and let ρ_1 and ρ_2 be respectively optimal T_1-prefix and T_2-prefix. To evaluate the score between ρ_1 and ρ_2 we consider two cases depending on ρ_1 and ρ_2 are or not empty:

(1) $\rho_1 = \emptyset$ and $\rho_2 = \emptyset$ are respectively both valid T_1-prefix and T_2-prefix, in this case:

$$S(\rho_1, \rho_2) = 0$$

(2) $\rho_1 \neq \emptyset$ and $\rho_2 \neq \emptyset$. If ρ_1 is empty then necessarily ρ_2 is empty and reciprocally. Then, during the edition of ρ_1 and ρ_2 and according to the three edit operations, we consider three cases:

(a) x_1 and x_2 has been substituted:

$$S(\rho_1, \rho_2) = LS(F_1[l(x_1) \dots x_1 - 1], F_2[l(x_2) \dots x_2 - 1]) + s(\alpha(x_1), \alpha(x_2))$$

(b) either x_1 has been deleted:

$$S(\rho_1, \rho_2) = LS(F_1[l(x_1) \dots x_1 - 1], T_2[x_2]) + s(\alpha(x_1), \lambda)$$

(c) x_2 has been inserted:

$$S(\rho_1, \rho_2) = LS(T_1[x_1], F_2[l(x_2) \dots x_2 - 1]) + s(\lambda, \alpha(x_2))$$

Therefore, the computation of the local score between two trees leads to evaluate local similarity between two forests.

4.3 Case of forests

Let $F_1[x_1 \dots y_1]$ and $F_2[x_2 \dots y_2]$ two forests and and let ρ_1 and ρ_2 be respectively optimal $F_1[x_1 \dots y_1]$-prefix and $F_2[x_2 \dots y_2]$-prefix. By definition ρ_1 and ρ_2 can be decomposed into prefixes of subtrees of $F_1[x_1 \dots y_1]$ and $F_2[x_2 \dots y_2]$, let τ_1 and τ_2 be respectively the prefix of $T_1[y_1]$ and $T_2[y_2]$. The computation

of the score between ρ_1 and ρ_2 can be decomposed into four cases depending on τ_1 and τ_2 are empty or not:

(1) $\tau_1 = \emptyset$ and $\tau_2 = \emptyset$ are respectively both valid $T_1[y_1]$-prefix and $T_2[y_2]$-prefix, in this case scores of deletion of $T_1[y_1]$ and insertion of $T_2[y_2]$ should not be taken into account:

$$S(\rho_1, \rho_2) = LS(F_1[x_1 \dots l(y_1) - 1], F_2[x_2 \dots l(y_2) - 1])$$

(2) $\tau_1 \neq \emptyset$ and $\tau_2 = \emptyset$, in this case the score of insertion of $T_2[y_2]$ should not be taken into account:

$$S(\rho_1, \rho_2) = LS(F_1[x_1 \dots y_1], F_2[x_2 \dots l(y_2) - 1])$$

(3) $\tau_1 = \emptyset$ and $\tau_2 \neq \emptyset$, in this case the score of deletion of $T_1[y_1]$ should not be taken into account:

$$S(\rho_1, \rho_2) = LS(F_1[x_1 \dots l(y_1) - 1], F_2[x_2 \dots y_2])$$

(4) $\tau_1 \neq \emptyset$ and $\tau_2 \neq \emptyset$, in this case the score of deletion of vertices of $T_1[y_1]$ and the score of insertion of vertices of $T_2[y_2]$ should be taken into account. Then, during the edition of ρ_1 and ρ_2 and according to the three edit operations, we consider three cases:
(a) y_1 and y_2 has been substituted:

$$S(\rho_1, \rho_2) = LS(F_1(x_1 \dots l(y_1) - 1], F_2[x_2 \dots l(y_2) - 1])$$

$$+LS(F_1(l(y_1) \dots y_1 - 1], F_2[l(y_2) \dots y_2 - 1]) + s(\alpha(y_1), \alpha(y_2))$$

(b) either y_1 has been deleted:

$$S(\rho_1, \rho_2) = LS(F_1[x_1, y_1 - 1], F_2[x_2 \dots y_2]) + s(\alpha(y_1), \lambda)$$

(c) y_2 has been inserted:

$$S(\rho_1, \rho_2) = LS(F_1[x_1 \dots y_1], F_2[x_2 \dots x_2 - 1]) + s(\lambda, \alpha(x_2))$$

The recurrence formula for the computation of the local score is thus given by the following proposition:

Proposition 5 *Let T_1 and T_2 be two trees and let x_1, y_1 and x_2, y_2 be vertices of T_1 and T_2 respectively, with $x_1 < y_1$ and $x_2 < y_2$:*

$$LS(F_1[x_1 \dots y_1], F_2[x_2 \dots y_2]) = \max \begin{cases} LS(F_1[x_1 \dots l(y_1)-1], F_2[x_2 \dots l(y_2)-1]) \\ LS(F_1[x_1 \dots y_1], F_2[x_2 \dots l(y_2)-1]) \\ LS(F_1[x_1 \dots l(y_1)-1], F_2[x_2 \dots y_2]) \\ LS(F_1[x_1 \dots l(y_1)-1], F_2[x_2 \dots l(y_2)-1]) \\ \quad +LS(F_1(l(y_1) \dots y_1-1], F_2[l(y_2) \dots y_2-1]) \\ \quad +s(\alpha(y_1), \alpha(y_2)) \\ LS(F_1[x_1 \dots y_1], F_2[x_2 \dots y_2-1]) + s(\lambda, \alpha(y_2)) \\ LS(F_1[x_1 \dots y_1-1], F_2[x_2 \dots y_2]) + s(\alpha(y_1), \lambda) \end{cases} \quad (3)$$

The complexity of this algorithm (given in Appendix A) is the same as Zhang-Shasha's algorithm and is bounded by $O(|T_1| \times |T_2| \times min(h(T_1), l(T_1)) \times min(h(T_2), l(T_2)))$ where, for any i in $\{1, 2\}$, $h(T_i)$ and $l(T_i)$ represent respectively the height and the number of leaves of the tree T_i. The average complexity of the algorithm is on the order of $|T_1|^{3/2} \times |T_2|^{3/2}$ ([19]). The space complexity is in $O(|T_1| \times |T_2|)$.

This recurrence formula is not totally equivalent to Smith and Waterman's computation [16]. Let us consider the problem of local similarity between sequences as a tree edit problem. Any sequence with a length n could be represented following two different graphs, *ie.* as a graph with n vertices and such that any vertex has only one child except one, the leaf, or as a graph made of a root and exactly $n-1$ children. In the first case, since any vertex (except the leaf) has only one child, the local score $LS(F_1[x_1 \dots l(y_1)-1], F_2[x_2 \dots l(y_2)-1])$ is always equal to zero. Similarly, three first line of equation 3 are equivalent to zero. Then the computation of local similarity between trees leads to the same equivalence relation than Smith and Waterman's one [16]. However, since order relations are not taken into account to define the notion of optimal T-prefix, the second case does not lead to the same result. Then to get Smith and Waterman result, a sequence of symbol should be represented using the first model.

4.4 Local comparison between quotiented trees

We consider in this section the generalisation of proposition 5 to quotiented trees. Let $Q_1 = (T_1, W_1, \pi_1)$ and $Q_2 = (T_2, W_2, \pi_2)$ be two quotiented trees. Since definitions of local prefixes and local scores presented in previous subsection are independant of the edit score, the local comparison bewteen quotiented trees consists in computing the local similarity between quotient trees $\pi(T_1)$ and $\pi(T_2)$ using the support subtrees to compute the scores.

Thus, from proposition 5, local score between quotiented trees can be recursively computed as follow:

Proposition 6 *Let $Q_1 = (T_1, W_1, \pi)$ and $Q_2 = (T_2, W_2, \pi)$ be two ordered quotiented trees and X_1, X_2, Y_1 and Y_2 four quotient vertices of $\pi(T_1)$ and*

$\pi(T_2)$ *($(X_1, X_2) \in W_1 \times W_1$, $X_1 \leq X_2$ and $(Y_1, Y_2) \in W_2 \times W_2$, $Y_1 \leq Y_2$)*

$$LS_Q(Q_1[X_1 \dots Y_1], Q_2[X_2 \dots Y_2]) = \max \begin{cases} LS_Q(Q_1[X_1 \dots l(Y_1) - 1], Q_2[X_2 \dots l(Y_2) - 1]) \\ LS_Q(Q_1[X_1 \dots Y_1], Q_2[X_2 \dots l(Y_2) - 1]) \\ LS_Q(Q_1[X_1 \dots l(Y_1) - 1], Q_2[X_2 \dots Y_2]) \\ LS_Q(Q_1[X_1 \dots l(Y_1) - 1], Q_2[X_2 \dots l(Y_2) - 1]) \\ \quad + LS_Q(Q_1[l(Y_1) \dots Y_1 - 1], Q_2[l(Y_2) \dots Y_2 - 1]) \\ \quad + S(\pi^{-1}(Y_1), \pi^{-1}(Y_2)) \\ LS_Q(\pi(T_1)[Y_1], Q_2[X_2 \dots Y_2 - 1]) + S(\theta, \pi^{-1}(Y_2)) \\ LS_Q(Q_1[X_1 \dots Y_1 - 1], \pi(T_2)[Y_2]) + S(\pi^{-1}(Y_1), \theta) \end{cases}$$

Considering the variables previously defined, the complexity of the algorithm is bounded by $O(|W_1| \times \min(l(\pi(T_1)), h(\pi(T_1))) \times |W_2| \times \min(l(\pi(T_2)), h(\pi(T_2))) \times \max_{t_1 \in W_1} \{|t_1| \times \min(h(t_1), l(t_1))\} \times \max_{t_2 \in W_2} \{|t_2| \times \min(h(t_2), l(t_2))\}$ where $h(t_i)$ and $l(t_i)$ are the height and the number of leaves of the tree t_i. The space complexity is in $O(|T_1| \times |T_2|)$.

5 Biological considerations

This section briefly illustrates the use of the global and local comparison methods in application context.

After a piece-by-piece comparison, algorithms provide the optimal sequence of edit operations. The user can thus identify which vertices are substituted during the comparison. Thus, the comparison method gives the user a qualitative outline of the similar subparts of both trees that are conserved. For instance, figure 4 represents the results obtained from a comparison of two quotiented trees using a global (Fig. 4a.) and a local (Fig. 4b) approach with a score of 1 for a substitution of a symbol by himself, and -1 for any other edit operation (insertion, deletion or substitution of a symbol by another one). Vertices conserved by substitutions are represented in black color. We can note that during the comparison conserved regions are distributed in several small connected groups whereas the local quotiented algorithm gathers these small mappings into a more dense one.

These approaches have been currently implemented in the AMAPmod system [23], a software originally dedicated to plant architecture analysis and more generally to analyze any object represented as tree-graphs. The methodology is currently being validated on RNase P RNA secondary structures of prokaryotes as well as on SSU and LSU ribosomal RNA. The detailed analysis of their structure organization has been carried out.

Descriptions of RNAs commonly rely on a tree graph representation [3,24]. A RNA secondary structure can be represented by a tree where vertices labels are:

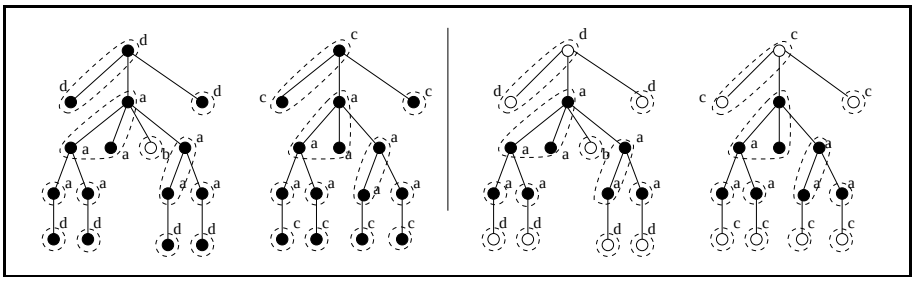

Fig. 4. Global (left) and local (right) quotiented edition of trees. Vertices that are substituted during the edition are in black, vertices inserted or deleted are represented in white.

- structural elements (sequences of paired bases, hairpins, loop, bulges, stems).
- or nucleic acids (A, C, G, U) and pairs of nucleic acids.

Both tree models of a RNA secondary structure represent information in the molecule at two distinct scales and are gathered in a single model: a quotiented tree.

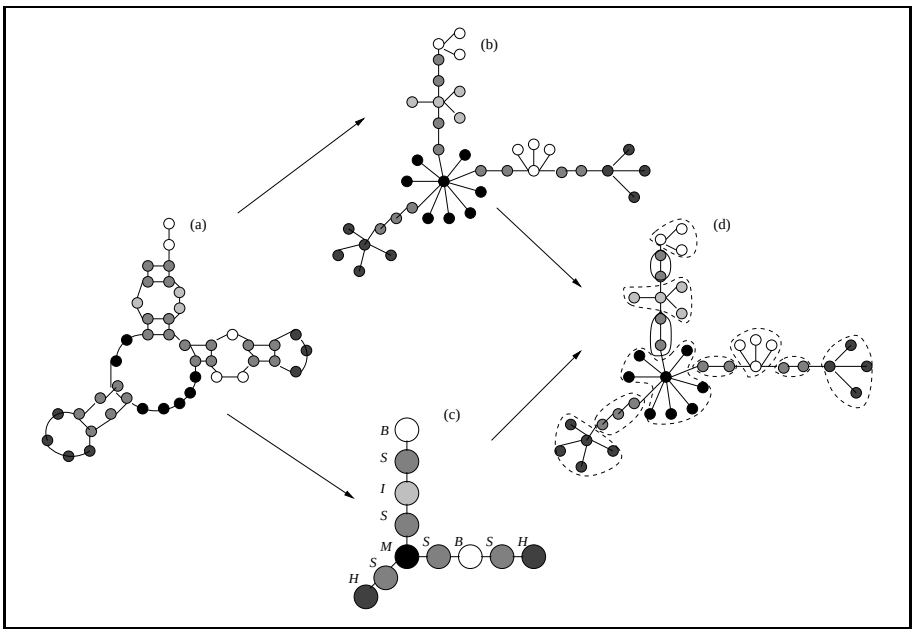

Fig. 5. The RNA secondary structure (a) is modeled by a microscopic tree (b) and a macroscopic one (c). Both trees are gathered to obtain the quotiented tree representation (d)

RNA secondary structures used to be compared using either microscopic [25] or macroscopic [3] tree representations. A quotiented comparison will take into account structural informations at both scales. Figure 6 shows an early example of global and local edition algorithms on quotiented tree representation of RNase P RNA secondary structure of Chlamydia Trachomatis and Halobacterium Cutirubrum. We can establish on this example that our algorithmic approach avoid the dispersion of paired bases, thus merging the conserved areas (substituted parts appear in darker characters).

6 Conclusion

In this paper we have extended an algorithm to compute distance between ordered trees [1] in order to define a method to globally and locally compare quotiented trees. These algorithms allow to consider two levels of details within the trees and take into account the structural elements of the trees. Resulting algorithms compute the optimal score recursively in polynomial time, using the dynamic programming principle. The final complexity has the same complexity than Zhang and Shasha [1] algorithm.

Works presented here are part of a project to develop a set of tools for analyzing biological objects modeled by rooted tree graphs [13]. Proposed algorithms and their implementation are currently integrated into this tool set.

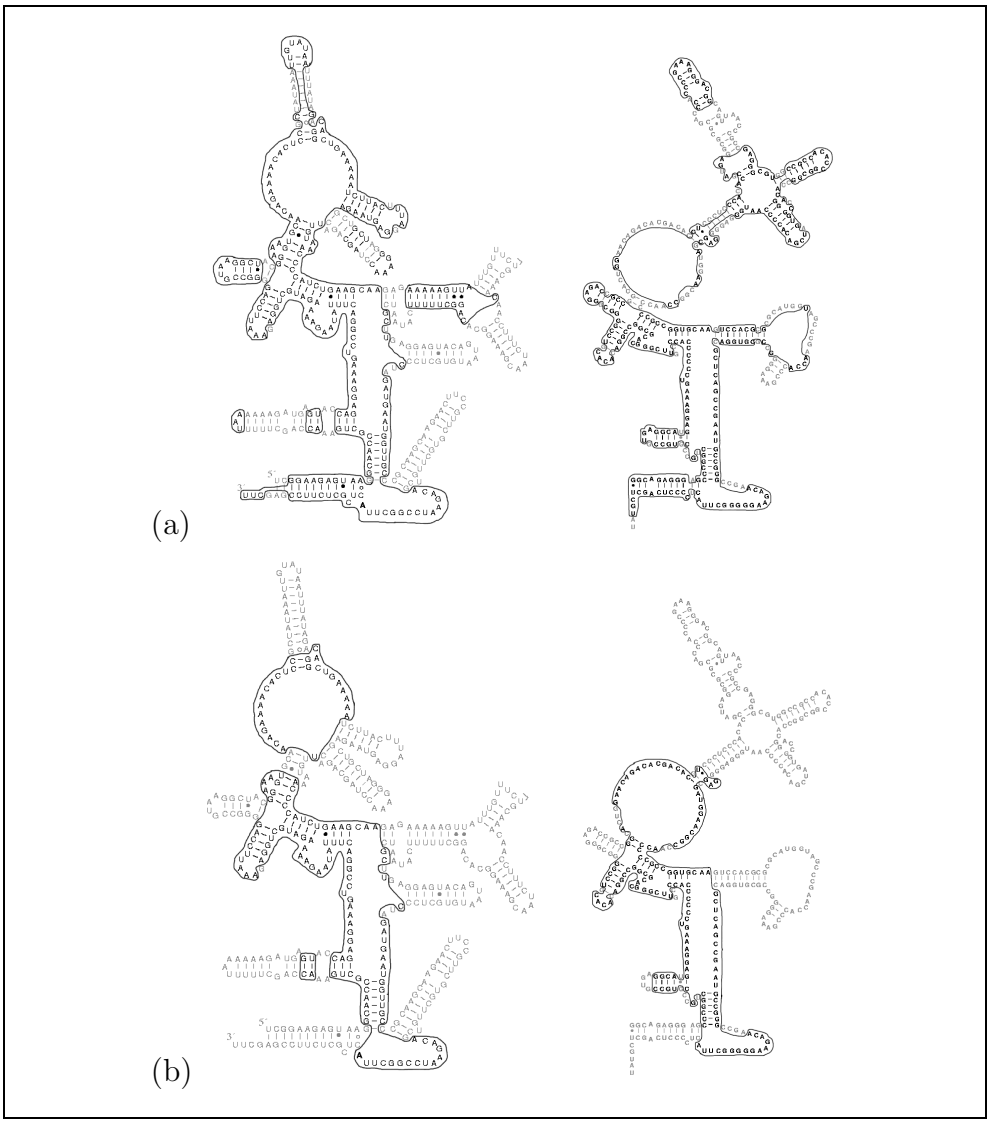

Fig. 6. (a) Global and (b) local quotiented edition of Chlamydia Trachomatis and Halobacterium Cutirubrum.

A Algorithm

Algorithm 1 *Local score between ordered trees*

$LS_{Max} = 0$; $LS(\theta,\theta) = 0$; $LS(\theta,\theta) = 0$; $M = \emptyset$;
-- LS and LS are two matrices of local scores between trees and forests indexed by vertices of T_1 and T_2.
-- $KeyRoots(T_1)$ and $KeyRoots(T_2)$ are the roots of the special subtrees of T_1 and T_2 respectively
For v **in** T_1 **Do** $LS(T_1[v],\theta) \leftarrow 0$
For w **in** T_2 **Do** $LS(\theta,T_2[j]) \leftarrow 0$
For v **in** $KeyRoots(T_1)$ **Do**
  For $i = l(v)$ **to** v **Do**
    $LS(F_1[l(v)\dots i],\theta) \leftarrow 0$
For w **in** $KeyRoots(T_2)$ **Do**
  For $j = l(w)$ **to** w **Do**
    $LS(\theta,F_2[l(w)\dots j]) \leftarrow 0$
For v **in** $KeyRoots(T_1)$ **Do**
  For w **in** $KeyRoots(T_2)$ **Do**
    For $i = l(v)$ **to** v **Do**
      For $j = l(w)$ **to** w **Do**
        $LS \leftarrow 0$
        If $l(i) = l(v)$ **and** $l(j) = l(w)$ **Then**
          If $LS < LS(F_1[l(v)\dots i-1],F_2[l(w)\dots j-1]) + s(i,j)$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots i-1],F_2[l(w)\dots j-1]) + s(i,j)$
            $case \leftarrow 1$
          If $LS < LS(F_1[l(v)\dots i-1],F_2[l(w)\dots j]) + s(i,\lambda)$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots i-1],F_2[l(w)\dots j]) + s(i,\lambda)$
            $case \leftarrow 2$
          If $LS < LS(F_1[l(v)\dots i],F_2[l(w)\dots j-1]) + s(\lambda,j)$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots i],F_2[l(w)\dots j-1]) + s(\lambda,j)$
            $case \leftarrow 3$
          $LS(T_1[v],T_2[w]) \leftarrow LS$
        Else
          $LS \leftarrow 0$
          If $LS < LS(F_1[l(v)\dots l(i)-1],F_2[l(w)\dots l(j)-1])$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots l(i)-1],F_2[l(w)\dots l(j)-1])$
            $caseF \leftarrow 1$
          If $LS < LS(F_1[l(v)\dots i],F_2[l(w)\dots l(j)-1])$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots i],F_2[l(w)\dots l(j)-1])$
            $caseF \leftarrow 2$
          If $LS < LS(F_1[l(v)\dots l(i)-1],F_2[l(w)\dots j])$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots l(i)-1],F_2[l(w)\dots j])$
            $caseF \leftarrow 3$
          If $LS < LS(F_1[l(v)\dots l(i)-1],F_2[l(w)\dots l(j)-1])$
          $+LS(F_1[l(i)\dots i-1],F_2[l(j)\dots j-1]) + s(i,j)$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots l(i)-1],F_2[l(w)\dots l(j)-1])$
            $+LS(F_1[l(i)\dots i-1],F_2[l(j)\dots j-1]) + s(i,j)$
            $caseF \leftarrow 4$
          If $LS < LS(F_1[l(v)\dots i],F_2[l(w)\dots j-1]) + s(\lambda,j)$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots i],F_2[l(w)\dots j-1]) + s(\lambda,j)$
            $caseF \leftarrow 5$
          If $LS < LS(F_1[l(v)\dots i-1],F_2[l(w)\dots j]) + s(i,\lambda)$ **Then**
            $LS \leftarrow LS(F_1[l(v)\dots i-1],F_2[l(w)\dots j]) + s(i,\lambda)$
            $caseF \leftarrow 3$
        $LS(F_1[l(v)\dots i],F_2[l(w)\dots j]) \leftarrow LS$
-- We store the best local prefix tree
        If $LS > LS_{Max}$ **Then**
          $LS_{Max} \leftarrow LS$ **and** $(v_{Max},w_{Max}) \leftarrow (v,w)$
-- Computation of the Mapping List
        If $case = 1$ **Then**
          $M(i,j) \leftarrow FM(i,j) \cup \{(i,j)\}$
          $FM(v,w) \leftarrow FM(v,w) \cup M(i,j)$
        Else If $caseF = 4$ **Then**
          $FM(v,w) \leftarrow FM(v,w) \cup M(i,j)$

return LS_{Max} *and* $M(v_{Max},w_{Max})$

References

[1] K. Zhang, D. Shasha, Simple fast algorithms for the editing distance between trees and related problems (1989) 1245–1262.

[2] G. Collins, S. Le, K. Zhang, A new algorithm for computing similarity between RNA structures, in: Proceedings of the 5th Joint Conference on Information Sciences, 2000, pp. 761–765.

[3] B. A. Shapiro, K. Zhang, Comparing multiple RNA secondary structures using tree comparisons, Cabios 6 (1990) 309–318.

[4] P. Ferraro, C. Godin, A distance measure between plant architectures, Annals of Forest Science 57 (2000) 445–461.

[5] S. Lu, K. FU, Error-correcting tree automata for syntactic pattern recognition, IEEE Trans. Computers 27 (1978) 1040–1053.

[6] R. A. Wagner, M. J. Fisher, The string-to-string correction problem, Journal of the association for computing machinery 21 (1974) 168–173.

[7] S. M. Selkow, The tree-to-tree editing problem, Information processing letters (1977) 184–186.

[8] K.-C. Tai, The tree-to-tree correction problem, Journal of the Association for Computing Machinery (1979) 422–433.

[9] K. Zhang, A new editing-based distance between unordered trees, in: Combinatorial Pattern Matching, 4th Ann. Symp., CPM'93, Padala (Italy), 1993, pp. 254–265.

[10] P. Ferraro, Méthodes algorithmiques de comparaison d'arborescences. Applications à la comparaison de l'architecture des plantes, Phd thesis, Institut National Polytechnique de Toulouse, France (November 2000).

[11] E. Tanaka, K. Tanaka, The tree-to-tree editing problem, International Journal Pattern Recognition and Artificial Intelligency 2 (2) (1988) 221–240.

[12] T. Jiang, L. Wang, K. Zhang, Alignment of trees - an alternative to tree edit, in: Combinatorial Pattern Matching'94, 5th Annual Symposium, 1994, pp. 75–86.

[13] C. Godin, Y. Caraglio, A multiscale model of plant topological structures, Journal of theoretical biology 191 (1998) 1–46.

[14] A. Ouangraoua, Distance multi-échelles entre structures secondaires d'ARN, Tech. rep., Université Bordeaux 1 - LaBRI (2004).

[15] P. Ferraro, C. Godin, An edit distance between quotiented trees, Algorithmica 36 (2003) 1–39.

[16] T. Smith, M. Waterman, Identification of common molecular subsequences, Journal of Molecular Biology 147 (1981) 195–197.

[17] M. Vauchaussade, X. Viennot, Enumeration of RNA secondary structures by complexity (1985) 360–365.

[18] W. Schmitt, M. Waterman, Linear trees and RNA secondary structures (1994) 317–323.

[19] S. Dulucq, L. Tichit, RNA secondary structure comparison: exact analysis of the Zhang-Shasha tree edit algorithm 306 (2003) 471–484.

[20] M. Höchsmann, T. Töller, R. Giegerich, S. Kurtz, Local similarity in RNA secondary structures, in: Proceedings of Computational Systems Bioinformatics, (CSB'03), 2003, pp. 159–168.

[21] J. T. Wang, B. A. Shapiro, D. Shasha, K. Zhang, K. M. Currey, An algorithm for finding the largest approximately common substructures of two trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 889–895.

[22] J. T. Wang, K. Zhang, C. Y. Chang, Identifying approximately common substructures in trees based on a restricted edit distance, Information Sciences 121 (1999) 367–386.

[23] C. Godin, Y. Guédon, E. Costes, Y. Caraglio, Measuring and analyzing plants with the amapmod software, in: M. Michalewicz (Ed.), Advances in computational life sciences, Vol I : Plants to ecosystems, Vol. January, Csiro, Australia, 1997, pp. 63–94, chapitre 4.

[24] M. Vauchassade De Chaumont, X. Viennot, Polynômes orthogonaux et problèmes d'énumération en biologie moléculaire, in: Séminaire Lotharingien de Combinatoire, B08l, 1984, pp. 79–86.

[25] K. Zhang, Computing similarity between RNA seondary structures, in: Proceedings of IEEE International Joint Symposia on Intelligence and Systems, Rockville, Maryland, 1998, pp. 126–132.